Experimental Verification of Discrete Switching Vibration Suppression

Joshua Schultz and Jun Ueda, *Member, IEEE*

***Abstract*—Control system design for flexible robotic systems requires special care with regard to the control system design to prevent oscillation in the system's resonant modes. If the resonant frequencies of such a system are known, it is possible to determine a switching command that delivers comparable actuation without exciting these natural modes of vibration. If there is redundancy in actuation, it can be exploited to suppress vibration with a reduced amount of actuator changes in state. Minimum Switching Discrete Switching Vibration Suppression (MSDSVS) involves choosing a switching function with integer amplitudes and continuously variable switch timings to force the root of the residual oscillation function with respect to frequency to be at a resonance. By minimizing the 1 norm of the vector of amplitudes, we obtain several desired properties. Such a vibration suppression command is developed for a flexible robotic actuator, and experimental results are presented. The proposed command reduces residual oscillation by 73% (RMS) and 74% (largest Fourier component) and represents a 37% energy savings over vibration suppression commands that do not exploit the redundancy in actuation.**

I. Introduction

In recent years, robotics and automatic machinery have undergone a transition in the public consciousness. Once they were exclusively the domain of heavy manufacturing installations; now there exist several commercial products that perform familiar tasks. The robot's workspace is no longer cordoned off from human beings and as such, robots must be able to interact with humans safely. Higher manufacturing volumes of robotic products demand that material be conserved and less costly materials be used wherever possible. Industrial design has become more of a factor in the decision to purchase a robot. Because of these phenomena, robotic links have become thinner, more compliant and lighter, to the point where the mass of the manipulator is barely larger than that of the object being manipulated. For these reasons, the hardware designer is often no longer able to prevent oscillation in the members of robotic devices simply by making large, inflexible links, and the design teams rely more heavily on the control designer to achieve a suitably damped response in the presence of flexibility.

Piezoelectric actuators have received considerable attention in Mechatronics literature as of late. They also have been deployed in a number of commercial applications, such as auto focus mechanisms for cameras. By itself, a piezoelectric stack actuator has a very small displacement. One method of mitigating this is to place the piezoelectric stack in combination with a stick-slip friction mechanism to move some other part, as in [1]. Another tactic is to bond the piezoelectric material to a flexible structure, which creates robotic actuators with zero backlash and displacements suitable for robotic applications. However, the flexible structure causes these actuators to have lightly damped resonant modes that, unless suppressed, greatly compromise their usefulness. Grossard, et. al, present an elegant method to design such actuators, using a genetic algorithm, that ensures all such modes are collocated [2]. Liaw and Shirinzadeh [3] present control of a flexure-based micromanipulator mechanism using a neural network. In this paper, we develop a method to suppress vibration in robotic devices where the actuator of a robotic device has flexibility and demonstrate the method on a piezoelectric actuator with a flexible mechanism.

Approaches to limiting the amplitude of vibration of robotic members have been numerous, most of which augment the rigid body state space with beam deflection states. Various control laws have been proposed, e.g., traditional linear loop shaping, feedforward, and nonlinear control techniques, such as sliding mode control [4], and dynamic feedback linearization involving the first eigenfunction of an Euler-Bernoulli beam [5]. Book [6] enumerates many of the challenges common to control of flexible bodies and summarizes several approaches to modeling and mitigating the flexibility. Nenchev, et. al. [7] determined control laws for manipulators with certain kinds of kinematic redundancies. These control laws are able to suppress vibration of an elastic base by projecting the control effort into what they term the reaction null space.

In certain situations where the most significant natural frequencies of vibration are known, command shaping techniques, in which the reference command is modified outside of the feedback loop, can insure that the command does not excite any of the known modes of the system. Input shaping [8] is an approach where the reference signal is convolved with a series of impulses, whose spacing is determined by the resonant frequencies of the system. It produces vibration-free point to point motion at the expense of slowing the response. Since an input shaper's transfer function is outside the feedback loop, it cannot produce instability. Díaz, Perira, Feliu, and Cela provide a method to concurrently determine a shaper and add mass at strategic locations that minimizes the additional delay incurred by the shaper [9].

Pao [10] extended the input shaping concept to systems with multiple actuators. She showed that by considering all control inputs simultaneously, one can solve for a shaper that has a faster response than shaping each input separately. Lim, Stevens and How [11] take a convex optimization approach to

Joshua Schultz and Jun Ueda are with the George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0405, USA. Tel: 404-385-2763, Fax:404-894-8496, E-mail: joshua.schultz@gatech.edu, jun.ueda@me.gatech.edu.

extend the abilities of input shaping for multiple actuators to include additional effects, including reducing transient oscillation.

The robots used in [10] and [11] both have servomotors which are capable of implementing a continuously variable control input, but there are many systems where on-off actuators are capable of achieving the desired end effector position. In fact, in some cases switching control has certain advantages, such as simplifying the control interface and creating less waste heat, even when the actuators are capable of continuously variable actuation. In [12], the authors develop a direct switching controller for a servomotor current control loop that compares favorably with PWM and linear methods. This method can be extended to the full control loop, resulting in a form equivalent to sliding mode control. Barth and Goldfarb [13] develop a switching control law for pneumatic actuators that is also based on sliding mode control. Singhose, Seering, and Singer [14] showed that using a vector diagram approach with an on-off actuator can provide a vibration-free command in the same manner as input shaping. Each time the actuator changes from one state to another (forward, off, reverse) its effort is associated with an impulse. Each impulse is represented by a phasor in the complex plane. The direction of the phasor, ϕ_j, is the angle corresponding to the time delay from the initial impulse, t_j, with respect to the natural period of the mode of vibration, T_n, (angular frequency ω_n) being considered.

$$\phi_j = 2\pi \frac{t_j}{T_n} = \omega_n t_j \tag{1}$$

If the vector sum of these impulses is zero, there will be no residual oscillation at frequency ω_n after the command is completed. Several works [15] [16]and [17] extend this method to account for and minimize fuel usage, which is particularly important for flexible spacecraft. Song, Buck and Agrawal [18] combine input shaping with pulse width modulation, rather than using a vector diagram approach.

In [19] we extend the vector diagram approach to systems with multiple on-off actuators that actuate along the same generalized coordinate. One option, which we call All On/ All Off Control, is to use the aforementioned vector diagram approach, solve for the command as if there were a single input, similarly to the process in [16], and apply this command identically to all inputs. However, this is merely a subset of the entirety of available commands, which we refer to as Discrete Switching Vibration Suppression (DSVS). One particular case of interest is when the minimum possible number of actuator input transitions that achieves the desired control objective is used, termed Minimum Switching Discrete Switching Vibration Suppression (MSDSVS). In this paper, we apply the command generation architecture developed in [19] experimentally to a single flexible actuator. In essence, the method proposed in this paper finds a middle ground between input shaping, which modifies a continuously variable command using fractional impulses, and on-off command generation where all impulse amplitudes are unity. This is useful for systems with multiple discrete levels of actuation, but which lack the ability to implement a continuously variable command.

II. Discrete Switching Vibration Suppression

Typically, in systems with on-off actuators, the input has a fixed amplitude. Although Singhose, Seering and Singer [14] make provision for impulse amplitudes that take different values, in their work, the variation in amplitude is used mainly to account for reduction in oscillation over time due to damping effects, undamped amplitudes being normalized to one. If a degree of freedom is actuated by an array of actuators, all of which produce the same amount of generalized force, we can define the control input at any time to be

$$u(t) = \sum_{j=0}^{m-1} A_j \delta(t_j), A_j \in \mathbf{D} \subset \mathbb{Z}, 0 \notin \mathbf{D} \tag{2}$$

where A_j is some integer impulse amplitude, m is the total number of impulses, and $\delta(t_j)$ is the Dirac delta function. Each A_j is normalized with respect to the generalized force provided by a single actuator, which is assumed to have the same magnitude in either direction. Since applying an impulse j at some time corresponds to turning a certain number of actuators on or off, it is limited by practical considerations, namely, how many actuators are, at a given time, in the "on" state and therefore available to be turned off, and how many actuators are in the "off" state and are available to be turned on. Thus, the set of possible input amplitudes, $\mathbf{D}$, will change each time an impulse is applied.

The vibration due to impulse A_j can be represented as a multiplicity of phasors in the complex plane, each rotating at a natural frequency of vibration. We assume that any amplitude reduction due to damping is negligible on the time scale of interest. If damping is significant, appropriate scaling of the impulse amplitudes with time can be incorporated.

Because the impulse amplitudes must take integer values, we assume that a set of amplitude values are known, and that the times at which they are applied, which can vary continuously, are the unknowns. All timings can be expressed as a phase ϕ_j with respect to the lowest natural frequency of the system. In order to suppress vibration at all natural frequencies, the real and imaginary components of the residual oscillation at all natural frequencies must be zero. This condition can be derived from the vector diagram and amounts to solving the system below:

$$
\begin{aligned}
A_0 + A_1 \cos\phi_1 + \cdots + A_{m-1}\cos\phi_{m-1} &= 0 \\
A_1 \sin\phi_1 + \ldots + A_{m-1}\sin\phi_{m-1} &= 0 \\
A_0 + A_1 \cos\frac{f_2}{f_1}\phi_1 + \cdots + A_{m-1}\cos\frac{f_2}{f_1}\phi_{m-1} &= 0 \\
A_1 \sin\frac{f_2}{f_1}\phi_1 + \cdots + A_{m-1}\sin\frac{f_2}{f_1}\phi_{m-1} &= 0 \\
&\vdots \\
A_0 + A_1 \cos\frac{f_n}{f_1}\phi_1 + \cdots + A_{m-1}\cos\frac{f_n}{f_1}\phi_{m-1} &= 0 \\
A_1 \sin\frac{f_n}{f_1}\phi_1 + \cdots + A_{m-1}\sin\frac{f_n}{f_1}\phi_{m-1} &= 0
\end{aligned}
\tag{3}
$$

where f_i is the natural frequency of the ith mode and there are n significant modes. Since (3) contains two equations for every frequency to be suppressed, due to the real and imaginary parts, the command must contain 2 impulses occurring at some phase ϕ per frequency to satisfy the constraints.

Activating any single input in the array, while leaving the remaining inputs off, will result in a steady state displacement. This displacement will be the same regardless of which input is activated. We can normalize the steady state displacement due to any combination of inputs by this value. The desired steady state displacement can then be expressed as an integer goal displacement y_g, which corresponds to the total number of inputs turned on. This adds an additional constraint that must be satisfied:

$$\sum_{j=0}^{m-1} A_j = y_g \tag{4}$$

There may be many choices of A_j for which a set of ϕ_j exists that can achieve the goal displacement and suppress vibration at the frequencies considered. In order to achieve the minimum switching condition, we choose

$$\{A_j \in \mathbf{D} \subset \mathbb{Z}\} \text{ such that } \sum_{j}^{m-1} |A_j| \text{ is minimized} \tag{5}$$

III. Redundantly Actuated Flexible Cellular Actuator

When a device has multiple actuators for a single degree of freedom, we say that it possesses redundant actuation. The cellular actuator shown in Fig. 1 is one such device. It is constructed from commercially available strain-amplified [20] [21] [22] lead zirconate titanate (PZT) piezoelectric stack actuators from the CEDRAT corporation, model APA50XS. These have a maximum no load displacement of 80 μm [23]. We refer to this as the "first layer." Because each of the first layer actuators has such a small displacement, 6 of them are stacked in series so that the first layer output displacement is the sum of the 6 actuators. This strain is amplified by the "second layer," a rhomboidal high silicon bronze strain amplifier that surrounds the first layer. If a (non-square) rhombus has some

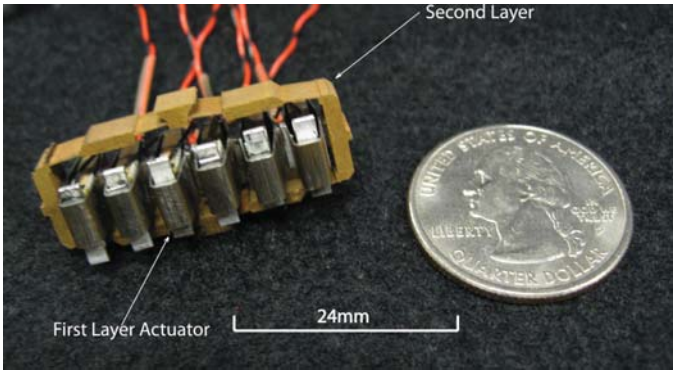


Fig. 1. Two layer cellular actuator

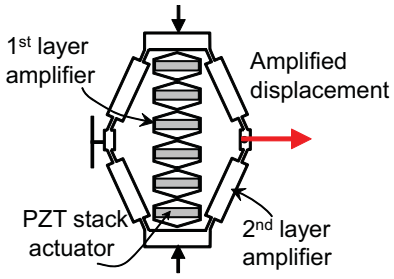


Fig. 2. Cellular Actuator Schematic

small strain along its major diagonal, ε_0, the minor diagonal will experience a larger strain, $\varepsilon_1 = \alpha_1 \varepsilon_0$ where $\alpha_1 > 1$ [21] [22]. The first layer units in series actuate along the major diagonal of the rhomboidal second layer. Fig. 2 illustrates this amplification concept. The second layer was designed to achieve 20% strain. The design process to achieve this displacement (along the minor diagonal) is described in detail in [22]. Cellular actuators can be cascaded to form robotic devices, as described in [24].

Because each first layer unit actuates in the same direction and has the same function, we say that the overall cellular actuator has redundant actuation. This type of actuator is equipped with redundancy a priori because several first layer units are necessary to achieve the desired stroke. Human muscle also possesses redundant actuation to increase stroke length and force. Any given muscle group will have many muscle fibers, or sarcomeres. Redundant actuation may also be present in traditional automation to insure the device can continue functioning if a single actuator fails.

This redundancy of actuation can be exploited to impose additional constraints on the system while achieving the desired control objective, e.g., position, force. Most notably, the redundancy of actuation can be used to limit vibration in the resonant modes of the actuator. Although the individual piezoelectric stacks are capable of receiving an analog command, it is desirable to restrict the command to each first layer actuator to be simply "on" or "off." The piezoelectric stacks possess hysteresis, which is problematic for positioning applications. On-off commands mitigate the hysteresis effect because the PZT stack only operates at the ends of the hysteresis loop [25]. Analog commands require transistors to operate in the linear

region, which complicates the drive electronics and causes more heat to be dissipated, requiring larger transistor spacing and increasing package size. Restricting the control input to switching commands simplifies the firmware. This quantization limits the position resolution, but when many actuators are combined in series-parallel combinations, the position is indistinguishable from that of a continuously variable analog command.

The cellular actuator has two significant modes ($n = 2$) that need to be suppressed. Because one end of the cellular actuator is fixed to a rigid support, the other end will have a finite relative displacement as the actuator extends. The goal displacement is normalized and can take any natural number from 0 to 6. In order for (3) and (4) to be satisfied, there must be a total of $m = 2n + 1 = 5$ impulses. $m > 5$ can be chosen if some strategy is chosen for determining a solution for an underconstrained system of equations, however, the additional impulses reduce the likelihood of reaching a minimum switching solution. Since the first impulse occurs at $t = 0$, the phases ϕ_j of the remaining four impulses must be found. Because the number of first layer units on cannot exceed six, the amplitudes A_j must be chosen such that

$$\{\forall j \in [0, 5] \subset \mathbb{Z}, A_j \mid 0 \leq \sum_{k=0}^{j} A_k \leq 6\} \tag{6}$$

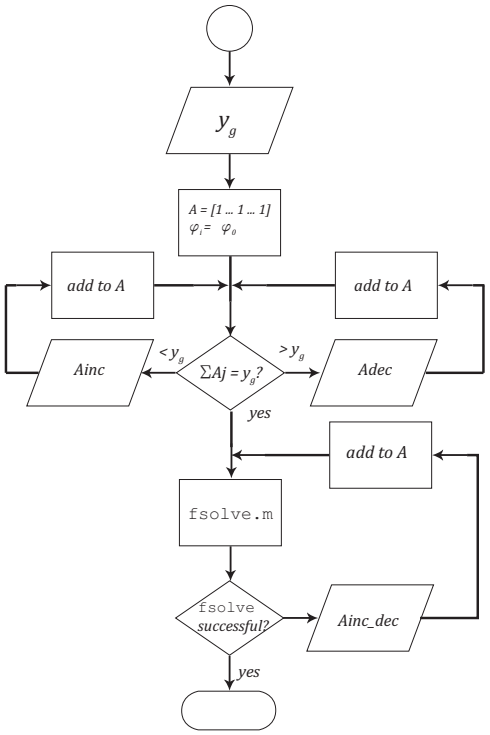


Fig. 3. Flowchart depicting algorithm to determine switching pattern

IV. Determination of Switching Pattern

A. General Case

The determination of the switching pattern for a general MSDSVS move is illustrated in Fig. 3 for the case where $m = 2n + 1$.

The algorithm breaks neatly into two parts: The first part determines a set of impulse amplitudes that will reach the goal displacement with the minimum number of changes in state to the inputs. The second part determines the phases that result in a vector sum of zero, or zero residual oscillation. $\mathbf{A}$ is initially set to a vector of length m, where all elements are 1. The vector of phases, ϕ is set to some convenient initial condition, such as evenly spaced angles between 0 and π. In general, the sum of all the elements of this intitial $\mathbf{A}$ will not equal the goal displacement, y_g. To correct this, the elements of $\mathbf{A}$ are either incremented or decremented using the rules shown in Table I. These rules ensure that no element of $\mathbf{A}$ will ever be zero, which would correspond to deleting an impulse. This would result in too few impulses to guarantee that (3) and (4) are satisfied. This results in a "stairstep" pattern (which contains, by inspection, the minimum number of switches) if $y_g >= m$. If $y_g < m$, this method produces one of several possible patterns with the minimum number of negative valued input transitions. Negative amplitudes correspond to inputs in the opposite direction of the desired displacement and represent wasted energy in one sense, however, they may be necessary to suppress all vibrational modes, especially for short moves. Once a "minimum switching" set of amplitudes is determined, the vector $\mathbf{A}$ is passed to a numerical solver which attempts to solve (3) by changing the phases ϕ_j. If the solver is successful, the set of amplitudes determined by the first part of the algorithm and the phases returned by the solver constitute the MDSDSVS command. If the solver is unable to determine a solution, $\mathbf{A}$ is modified using the final rule shown in Table I. This modifies the individual impulse amplitudes while leaving their sum unchanged, moving to the configuration with the next fewest number of individual input transitions.

B. Application to the Cellular Actuator

To generate commands for the cellular actuator, the phases ϕ_j are found by a custom MatLab function, `arb_novib2f.m`, which uses MatLab's (R2009b) native `fsolve.m` function. `fsolve.m` employs an algorithm based on that proposed by Powell in [26]. The cellular actuator has two significant resonant modes ($n = 2$), and 6 inputs, so 5 impulses are required for a minimum switching solution that achieves zero residual vibration. The algorithm begins with

$$\mathbf{A} = [A_0\ A_1\ \cdots\ A_5] = [1\ 1\ 1\ 1\ 1] \tag{7}$$

.

`arb_novib2f.m` then follows the procedure in the previous section to generate a sequence of impulses that results in a final position of y_g. $Ainc$ is some permutation of the elements of $[\ 0\ 0\ 0\ 1\ 0]^T$, $Adec$ is some permutation of the elements of $[\ 0\ \ -2\ 0\ 1\ 0]^T$. Once y_g is achieved, `fsolve.m` attempts to find Φ_j that result in zero residual vibration. If it is unsuccessful, $Ainc\_dec$ of the form $[\ 0\ -2\ 1\ 1\ 0]^T$ is added and `fsolve.m`. If it is possible to achieve a vibration-free move with fewer than 5 impulses, `fsolve.m` may calculate two identical Φ_j. In this case, the two impulses are removed and replaced by a single impulse whose amplitude is the sum

TABLE I
COMMAND SELECTION ALGORITHM RULES

Name	Form	Note
Ainc	$[\ 0\ \cdots\ 0\ 1\ 0\ \cdots\ 0\ ]^T$	
Adec	$[\ 0\ \cdots\ 0\ -1\ 0\ \cdots\ 0\ ]^T$	-1 has the same index as some $A_j \neq 1$
Adec	$[\ 0\ \cdots\ 0\ -2\ 0\ \cdots\ 0\ 2\ 0\ \cdots\ 0\ ]^T$	-2 has the same index as some $A_j = 1$
Adec	$[\ 0\ \cdots\ 0\ -2\ 0\ \cdots\ 0\ 1\ 0\ \cdots\ 0\ 1\ 0\ \cdots\ 0\ ]^T$	-2 has the same index as some $A_j = 1$
Ainc_dec	some Ainc + Adec	nonzero elements of Ainc and Adec may not have the same index

of the two amplitudes. An hypothetical example illustrating this process is provided in the appendix.

C. Existence and Uniqueness Considerations

If the first candidate set of amplitudes, $\mathbf{A}$ that satisfies (4) and (6) results in a successful set of ϕ_j that solves (3), for a given set of initial guesses, we know that it is a minimum switching solution. However, what if the `fsolve.m` fails to find a solution? We are certainly free to choose another set of initial conditions and try again, but simply trying many sets of initial guesses may still miss a valid solution. Even if the solver returns a valid solution, might it converge to a different solution with the same number of switches given a different set of initial guesses? Since (3) is a system on nonlinear equations, no statements of existence, non-existence, or uniqueness can be made using classical mathematical methods. Solutions to systems of nonlinear equations are still an area of active research.

Under certain conditions, the Interval Newton Method [27] may be used to make an analytical statement about the existence or uniqueness of a solution for a given set of amplitudes. This depends on the ability to invert the Jacobian of (3), which is singular at every multiple of $\frac{\pi}{4}\frac{f_k}{f_1}$ radians, where f_k is the highest natural frequency, so its application will apply to a small local region rather than globally. We intend to discuss the use of interval computations further in a forthcoming work. Because a solution may be near a singularity, small changes in the natural frequencies of the system may mean that the new solution is not in the neighborhood of the old solution. In fact, a candidate set of amplitudes may produce a solution for a given set of natural frequencies, and fail to produce a solution for neighboring frequencies. For this reason, it is important to verify all MSDSVS patterns experimentally.

V. EXPERIMENTAL SETUP

The relevant resonant frequencies to be suppressed and the desired final position (0-6 first units actuators on) are provided to `arb_novib2f.m`, which returns a table with the number of units on after each switch in the sequence and the time that each switch occurs. An on-off sequence is determined for each individual input that is feasible (a unit that is already on cannot be turned on), and the aggregate of all first layer units at any given time must match the command returned by `arb_novib2f.m`. If the choice of sequence specifies that each input must change state an odd number of times, the pattern for the forward direction can be used to command a motion of the length in the opposite direction, merely by inverting all input lines in the sequence. We call this a "reversible" pattern.

The DSVS command is implemented by a Silicon Laboratories C8051F120DK microcontroller development kit, running at 25.4 MHz. The command for each first layer unit is generated by each of the 6 Programmable Counter Array (PCA) modules, which toggles an output pin (CEX) state when a match occurs between a given module's internal register and the PCA clock, set to be the system clock/12. This allows implementation of a command which suppresses resonant frequencies as low as 32.3 Hz without rollover. The microcontroller is set to interrupt each time a PCA match occurs on any module, and the subsequent match times are loaded into the various PCA registers. If an output line is supposed to toggle, the PCA compare register is loaded with the next transition time at each match interrupt. If it is supposed to maintain its value at the next switching time, the PCA compare register is loaded with the maximum time value, preventing a match from occurring on that module when the next switching time is reached. When the final impulse occurs, the PCA clock is stopped and all output lines are held at their current values until the next move command occurs. Move commands are invoked by a serial command from a PC over RS-232 using the microcontroller's UART and PuTTY open-source telnet software.

Each microcontroller output is wired to an Avago ASSR302-C solid state switching IC, which has two bridges per chip. This is a commercially available, easily procured, low cost component, easily scalable to devices using a large number of cellular actuators. The "low" side of the PZT stack actuator in each first layer unit is always connected to ground. When the upper bridge is activated, the "high" side of the PZT is connected to ground. When the lower bridge is activated, the "high" side of the PZT stack is connected to the source voltage, imposing a voltage across the PZT stack and inducing a displacement. A pair of bridges is necessary because the PZT stack actuators are capacitive; simply disconnecting a stack actuator from the source voltage will not cause it to return to its unforced length. It must be grounded so that it can discharge. The ASSR302-C has a delay of 70μs and an R_{dson} of 30 Ω (typical). The drive voltage is provided by an American Power Design A5 series 5 Watt high voltage DC/DC converter. It supplies 33.3 mA at 150V. A delay circuit with a time delay of 40μs prevents cross conduction.

One end of the cellular actuator is fixed to a rigid support. The position of the other end is measured by a Micro-Epsilon optoNCDT ILD 2200-20 laser position sensor with a sampling frequency of 10 kHz. For the verification of the MSDSVS algorithm on cellular actuator hardware, the signal from the laser position sensor was recorded with a LeCroy Waverunner 44MXi mixed signal digital oscilloscope. A photo of the

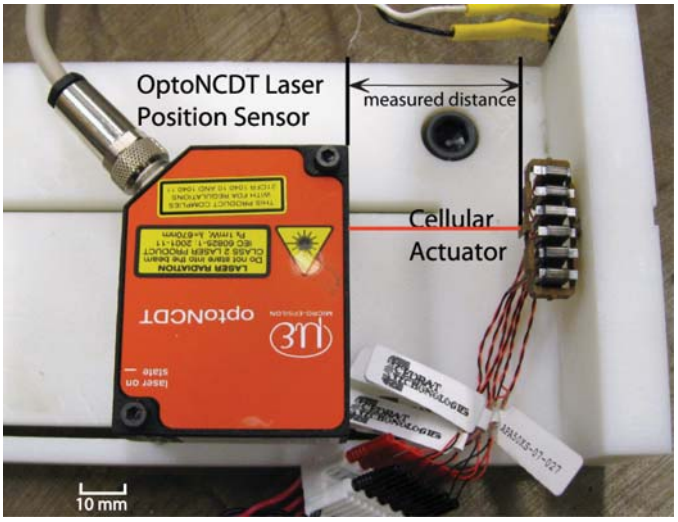


Fig. 4. Cellular Actuator Experimental Setup

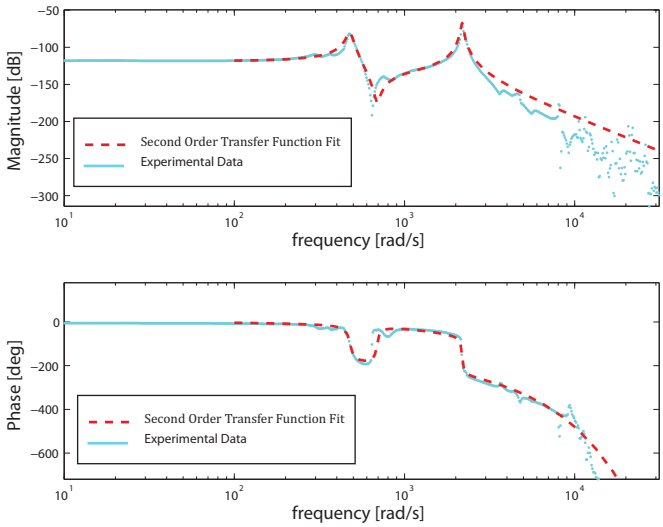


Fig. 5. Frequency Response of Cellular Actuator

TABLE II
PARAMETERS FOR EXPERIMENTAL FREQUENCY RESPONSE FIT

Mode	K	$\omega, [rad/s]$	ζ
1	1.46E-3	478.61	0.0400
2	1.22E-3	2180.3	0.0175

experimental setup is shown in Fig. 4.

VI. EXPERIMENTAL RESULTS

The frequency response of the cellular actuator was obtained using an HP 3562a spectrum analyzer to generate a sinusoidal input and compare it with the output of the laser position sensor. For the purpose of obtaining the frequency response, the cellular actuator was driven through a Cedrat CA45 amplifier, which has an unloaded bandwidth of 10 kHz. A transfer function for a general vibratory system with two significant modes of vibration with time delay was fit to the data. The frequency response is shown in Fig. 5. The resonant frequencies differ somewhat from those in [19]; the finite element analysis conducted in [19] used PZT material constants from a commonly available PZT material that differ from those of the Cedrat PZT material. In addition, some variation between finite element predictions and experimental values is to be expected in most cases.

Since the displacement generated by the actuator results from deformation only, and not from rigid body displacement, it can be represented as the sum of n lightly damped second order systems:

$$G(s) = \sum_{i=1}^{n} \frac{K_i}{s^2 + 2\zeta_i \omega_i s + \omega_i^2} \tag{8}$$

where for mode i, K_i is the residue, ω_i is the natural angular frequency, ζ is the damping ratio, and s is the LaPlace variable. The results from the experimental transfer function fit for the 2 most significant modes are in table II. The time delay in the system was determined from the fit to be 3.9 ms.

The MSDSVS and all on/all off commands (command for $y_g = 6$ shown in Fig. 6) were applied to the cellular actuator. Note that a monotonically increasing command was not found for this set of natural frequencies. The response to a step (all 6 actuators transitioned at the same time) and the response to both vibration suppression commands are shown in Fig. 7. The step response shows considerable oscillation (>50% overshoot), due to the lightly damped nature of the system. The various commands are completed by 10 ms, so the remaining period of time shows the decay of any residual oscillation once the command is completed. Note that the response to All On All Off command has the largest transient during the command.

Despite the use of vibration suppression commands, one still observes some oscillation remaining in the response after such a command is completed. This is likely due to a combination of factors. Both All On All Off and MSDSVS methods are based on linear analysis, and they suppress oscillation in an actuator that is described accurately by a linear model. In actuality, some non-negligible nonlinear effects are present. The natural frequency of the cellular actuator changes slightly as it extends. The values in Table II are only exact for the intermediate position corresponding to the DC offset of the spectrum analyzer, +2V, during the test because the frequency sweep with the spectrum analyzer caused oscillation about this intermediate position. The command was designed based on the frequency for this intermediate length, not the natural frequency of the actuator when it was extended. Most of this remaining oscillation could be removed by further empirical tuning if necessary. The second mode is particularly problematic because its frequency is high enough to where the dead time to prevent cross conduction and time delay in the system begins to become significant. The time constant of the PZT and switching circuit is beginning to become significant at the frequency of the second mode as well. For robotic devices using large numbers of cellular actuators, however, the dominant modes will not be high enough for this to be a concern. In addition, we have assumed that an “on” command has the same force magnitude as an “off” command, and

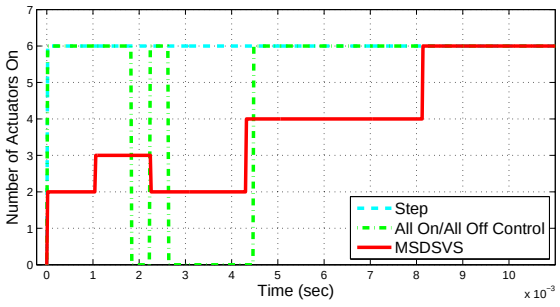


Fig. 6. Step, All On/All Off and MSDS Commands to Cellular Actuator, $y_g = 6$

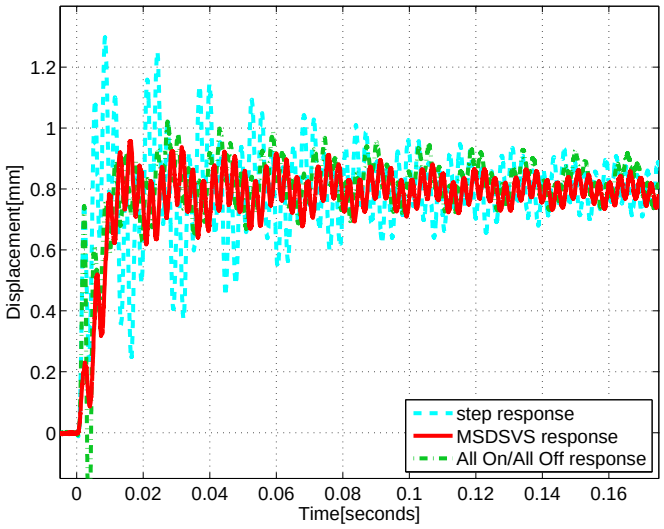


Fig. 7. Dynamic Response of Step, All On/All Off and MSDSVS Commands, $y_g = 6$

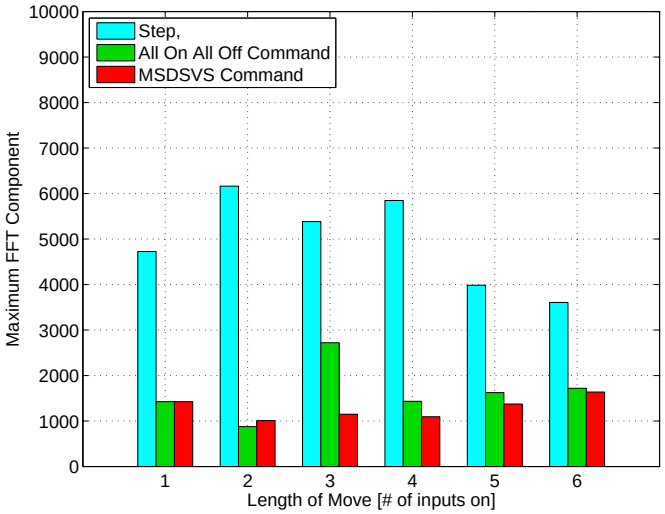


Fig. 8. Magnitude of Largest FFT Component, Normalized by Move Distance

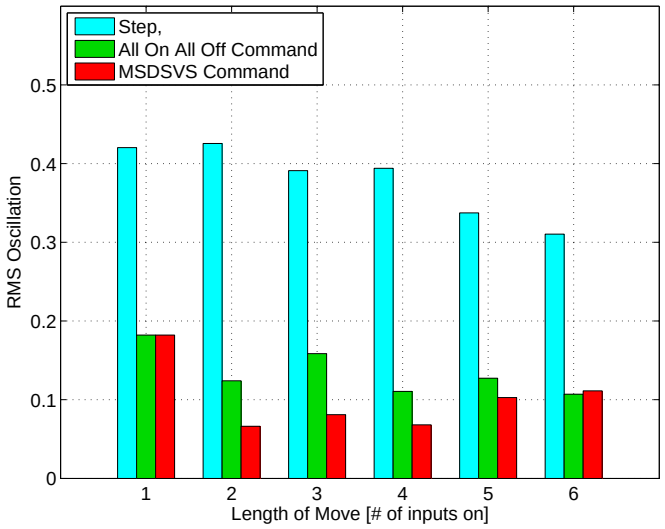


Fig. 9. RMS Oscillation, Normalized by Move Distance

this may not be entirely accurate for the cellular actuator. If these properties are well known, adjustments can be made to compensate for them when generating the command. Finally, during the frequency response testing, we observed that when the actuator was excited with a pure sinusoid at the frequency of the first resonant mode, the output was not a pure sinusoid. Instead, persistent oscillation at the frequency of the second mode was superimposed over the response at the excitation frequency, indicating a nonlinear dynamic effect which is not captured by the linear analysis.

When either MSDSVS or All On/All Off control is applied to the cellular actuator, the largest component of the FFT (Fig. 8) and the RMS oscillation (Fig. 9) are reduced by 50% or more in nearly all cases. In most cases, MSDSVS performed slightly better. For $y_g = 6$, the 2% settling time for a step command was measured to be 699 ms. For the all on all off command, the settling time was 477 ms. For MSDSVS, the settling time was 529 ms. With fine-tuning to compensate for all the non-ideal behaviors, we expect that the settling time for the vibration suppressed commands can be reduced greatly.

VII. Discussion

If both MSDSVS and All On/All Off control both reduce oscillation, the reader may wonder whether there are any advantages in using one over the other. The advantage of All On/All Off control is that the command is nearly always shorter in duration, and thus, in theory, it can reach a vibration-free response sooner than MSDSVS. However, there are a number of advantages to MSDSVS. MSDSVS, since it has fewer transitions, uses less energy than the All On/All Off solution. This corresponds to a smaller control effort being used. In this particular application, there is a certain amount of Joule heating that occurs when the PZT is energized and discharged. This can be expressed as

$$E_J = \int_0^T i^2 R_{dson}\, dt \tag{9}$$

where E_J is the energy dissipated, and i is the current, and

TABLE III
NUMBER OF SWITCHES REQUIRED TO REACH y_g FOR EACH COMMAND

y_g	Step	All On All Off	MSDSVS
1	1	5	5
2	2	10	6
3	3	15	5
4	4	20	8
5	5	25	7
6	6	30	8

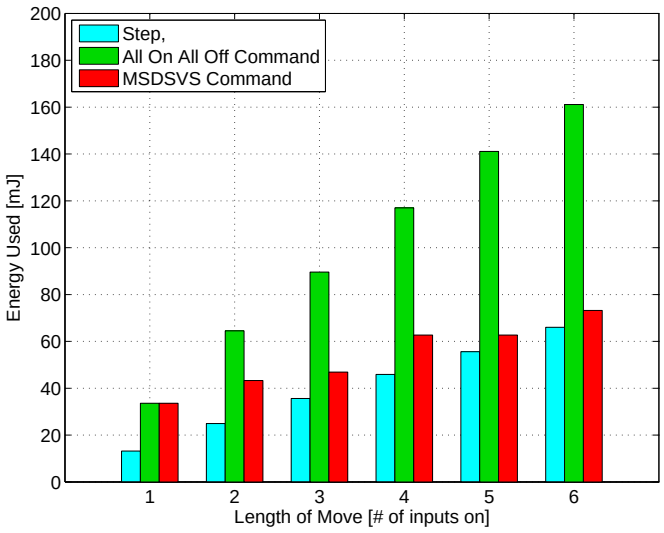


Fig. 10. Energy Consumption per Move

T is the charging/discharging time. In order to determine the amount of energy used for a move using a given command, we measured the voltage supplied to the input of the DC/DC converter using a LeCroy ADP305 differential probe. We measured the current to the DC-DC converter with a LeCroy AP015 current probe. After subtracting steady state leakage currents, the product of the two signals was integrated numerically over the time duration of the move. Fig. 10 shows how the difference in energy dissipated grows quickly with an increase in the number of inputs. Table III shows the number of switches required to reach a given position, which directly correlates to the results shown in Fig. 10. A "switch" corresponds to a transition of a single PZT stack, so, for instance, implementing an impulse of $A_j = 2$ would count as two "switches," since 2 stacks must be energized. Table IV summarizes the information in figures 8-10. The MSDSVS command reduces the RMS oscillation by 73% and the amplitude of the largest frequency component by 74%. All On All Off control reduces oscillation by nearly the same amount, but the MSDSVS command uses 37% less energy than the All On All Off Command and shows better robustness properties.

Minimizing the number of cycles also has benefits for the life of the actuator. Piezoelectric bimorphs have been known to delaminate at 10^8 cycles [25].

MSDSVS also has advantages with regard to robustness. Robustness is often characterized in terms of the residual amplitude of oscillation produced by a given All On All Off or MSDSVS command when an unmodeled or uncertain

TABLE IV
OVERALL PERFORMANCE COMPARISON BETWEEN COMMANDS
(RMS AND FFT ARE NORMALIZED)

Command	Avg RMS	Avg FFT	Avg Energy/Move Dist. [mJ]
Step	0.380	4950	11.9
All On All Off	0.135	1630	30.0
MSDSVS	0.102	1280	18.6

resonance of frequency ω is present.

$$A_R = \frac{|\sum_{k=0}^{m-1} A_k e^{\frac{j\omega\phi_k}{\omega_1}}|}{\sum_{k=0}^{m-1} A_k} \times 100\% \tag{10}$$

The sensitivity plot (A_r vs ω) of the MSDSVS command and the All On/All Off command is shown in Fig.11. MSDSVS shows much less amplification, in general, if there is an unmodeled lightly damped mode. Because of the periodic nature of the vector diagram summation, both types of commands will suppress additional frequencies higher than those they are designed to suppress. The All On All Off command has two such frequencies near 590 Hz and 720 Hz, but across the board, MSDSVS performs better. Fig. 12 shows the time response of the cellular actuator to an All On/All Off command and an MSDSVS command when both commands were designed with an assumed first mode natural frequency of 45.7 Hz, rather than the measured frequency of 76.1 Hz ($y_g = 6$). MSDSVS still exhibits fair vibration suppression properties, even when the frequency is off by 40%, whereas All On/All Off control exhibits almost no vibration suppression.

Figure 13 shows the response when two masses are added to the cellular actuator, but the commands are the same as shown in Fig 6, again producing a mismatch between the frequencies suppressed by the command and the natural frequency. The natural frequencies of the modified actuator are $f_1 = 56.8$ Hz and $f_2 = 341$ Hz. In this case, both All On All Off control and MSDSVS still provide some benefit. The amount of residual oscillation for the All On All off and MSDSVS methods are comparable, as predicted by Fig. 11 when the first mode is lowered.

VIII. CONCLUSION

This paper presents the discrete switching vibration suppression (DSVS) method for flexible robotic systems with redundancy in actuation. This represents a greater class of vibration suppression commands then applying existing approaches identically to all inputs. DSVS successfully reduces the amplitude of oscillation when applied to the redundant, flexible cellular actuator. Choosing the DSVS command with the minimum number of total input transitions has specific benefits, namely, lower energy usage and increased robustness to modeling errors. The cellular actuator is by no means unique, and this method could also be effective when applied to any flexible system that employs multiple on-off actuators in a single direction, such as a spacecraft with multiple jets that can fire independently. With some minor modifications in the assumptions, it can also be applied to a flexible system with highly discretized actuation, such as a flexible arm driven by a stepper motor with microstepping capability.

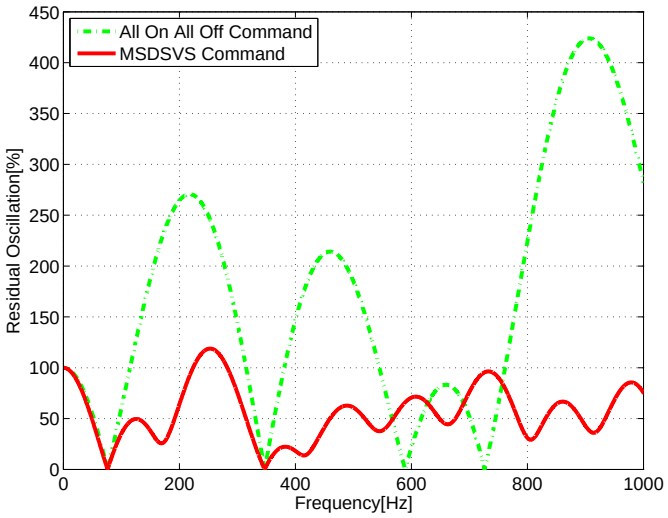


Fig. 11. Sensitivity Plot for All On/All Off Control and MSDSVS

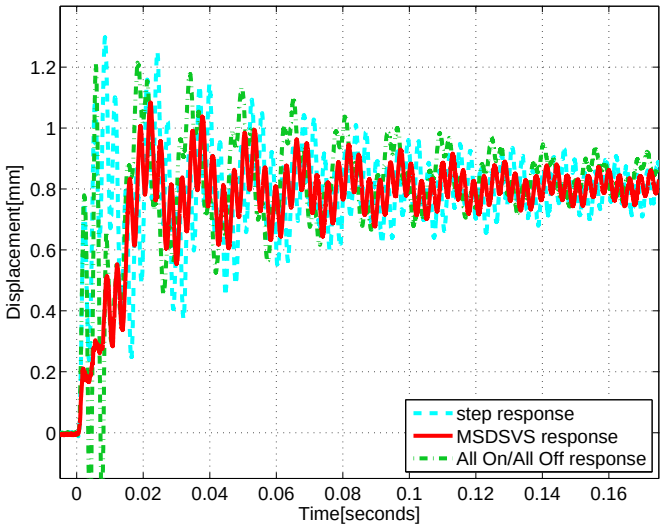


Fig. 12. Cellular Actuator Response When Command Designed Based on Incorrect Natural Frequency

Future work could include an analytical investigation of the existence and uniqueness of DSVS solutions for a given set of amplitudes, applications to systems of engineering interest driven by cellular actuators, a more efficient and exhaustive search algorithm, developing "robust" DSVS commands that use additional impulses to enforce a constraint on the first derivative of the residual oscillation with respect to frequency and extending the method to feedback based switching techniques which can eliminate oscillation from sources other than the command.

Appendix A
Cellular Actuator Mode Shapes

In [19] we used Dassault Systems' COSMOSWORKS to analyze the vibrational modes and predict natural frequencies and mode shapes. The mode shapes corresponding to the model shown in Fig. 5 are shown here.

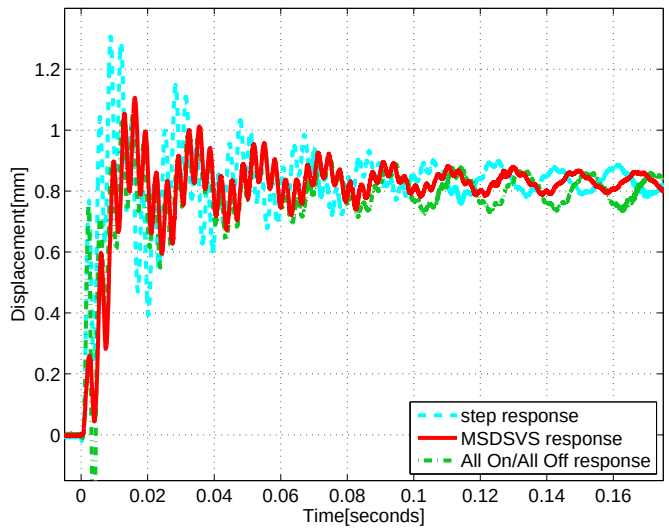
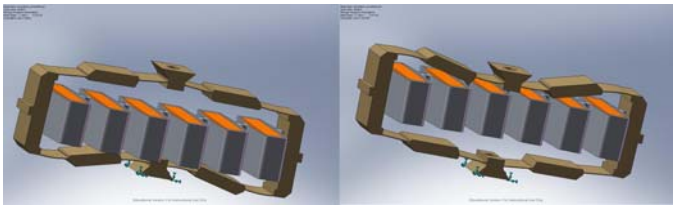


Fig. 13. Cellular Actuator Response when the Natural Frequency is Changed



left: mode 1, right: mode 2

Fig. 14. Principal Vibration Modes of Cellular Actuator

Appendix B
Commands for Additional Move Lengths

In addition to the full stroke $y_g = 6$, commands for the 5 intermediate positions were developed. All On All Off control and MSDSVS are identical for $y_g = 1$. They are shown here.

Appendix C
Illustrative Example of Switching Algorithm

Fig. 20 shows an example of the process conducted by `arb_novib_2f.m`. In this case, $y_g = 6$. The initial $\mathbf{A} = [\,1\;1\;1\;1\;1\,]$ results in a final position $\sum_{j=0}^{4} A_j$ of 5, not 6, so the pattern must be modified. Firstly, to satisfy (4), the vector $[\,0\;0\;0\;1\;0\,]$ is added to $\mathbf{A}$, causing the second to last impulse to have an amplitude of 2, which satisfies the desired y_g with a monotonically increasing command, so it is guaranteed to have the minimum number of switches. Next, `fsolve.m` attempts to find ϕ_j that satisfy (3). For purposes of illustration, let us say that in this case, `fsolve.m` does not return a valid solution. `fsolve.m` does not enforce $\phi_j < \phi_{j+1}$, so it is not strictly mathematically necessary to try different permutations of $\mathbf{A}$. However, from a numerical perspective, the presence of local minima and singularities may affect the reachability of a solution from a given set of initial conditions. We hope to develop some analytical tools to address this in a future work. Since the monotonic solution was not successful, we must introduce an amplitude of -1. In this example, we add

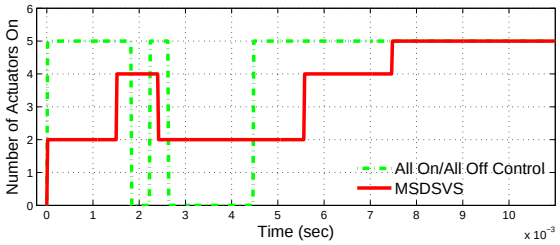


Fig. 15. $y_g = 5$

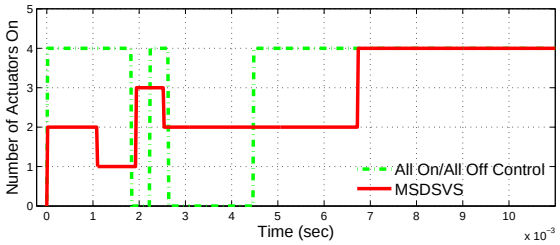


Fig. 16. $y_g = 4$

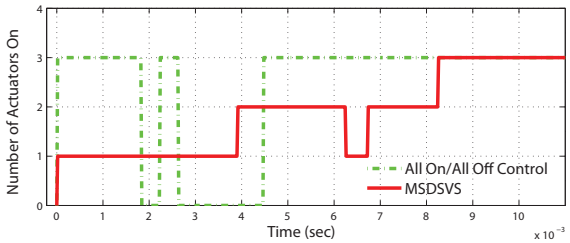


Fig. 17. $y_g = 3$

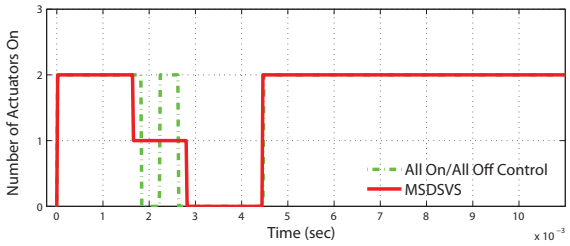


Fig. 18. $y_g = 2$

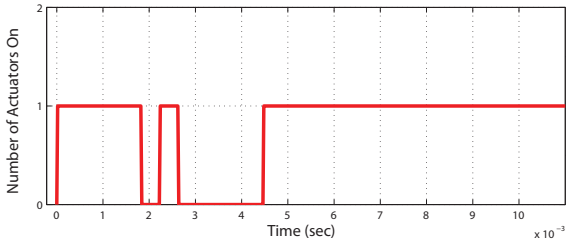


Fig. 19. $y_g = 1$

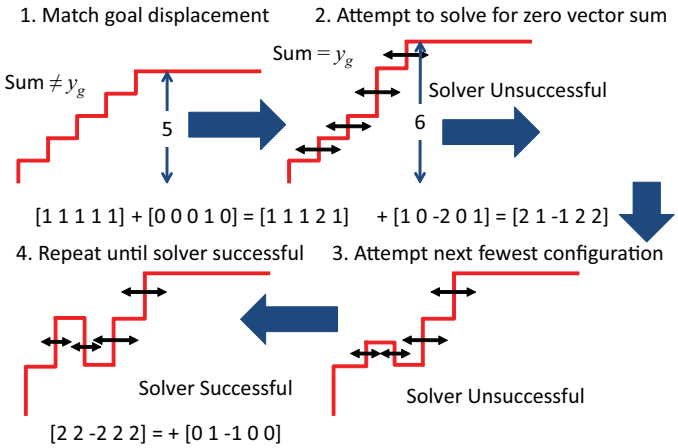


Fig. 20. Illustration of `arb_novib_2f.m` Operation

$[\ 1\ 0\ \ -2\ 0\ 1\ ]$ to $\mathbf{A}$. `fsolve.m` then attempts to find a solution. Let us say that it is once again unsuccessful. We than have the choice of introducing an additional negative impulse by repeating the previous step, or we can increase the absolute value of an existing negative impulse. In this example, we choose the latter, and add $[\ 0\ 1\ \ -1\ 0\ 0\ ]$. Let us say that this time `fsolve.m` returns a solution. This must be checked to make sure that it does not violate (6). When the A_j are arranged in order of ascending ϕ_j, this is determined to be the case. This is then designated as the minimum switching command.

To implement this command, at $t = 0$ two PZTs would be turned on, then two of those remaining off are turned on at $t = \frac{\phi_1}{\omega_1}$. At $t = \frac{\phi_2}{\omega_1}$ two of the four turned on at previous timings would be turned off. At this point there are two PZTs on and four off. Two of those off are turned on, and the remaining two turned on at the timings corresponding to ϕ_3 and ϕ_4 to complete the command. This results in a total of 10 "switches."

Acknowledgment

This research was supported by NSF grant Cyber-Physical Systems ECCS-0932208

References

[1] Y. Zhang, G. Liu, and J. Hesselbach, "On development of a rotary – linear actuator using piezoelectric translators," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 5, pp. 647 –650, oct. 2006.

[2] M. Grossard, C. Rotinat-Libersa, N. Chaillet, and M. Boukallel, "Mechanical and control-oriented design of a monolithic piezoelectric microgripper using a new topological optimization method," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 1, pp. 32 –45, feb. 2009.

[3] H. C. Liaw and B. Shirinzadeh, "Neural network motion tracking control of piezo-actuated flexure-based mechanisms for micro-/nanomanipulation," *Mechatronics, IEEE/ASME Transactions on*, vol. 14, no. 5, pp. 517 –527, oct. 2009.

[4] S. Choi, C. Cheong, and H. Shin, "Sliding mode control of vibration in a single-link flexible arm with parameter variations," *Journal of Sound and Vibration*, vol. 179, no. 5, pp. 737–748, 1995.

[5] A. D. Luca and G. D. Giovanni, "Rest-to rest motion of a two-link robot with a flexible forearm," in *Proceedings of ASME/IEEE International Conference on Advanced Intelligent Mechatronics*, vol. 2, 2001, pp. 929–935.

[6] W. Book, "Controlled motion in an elastic world," *Transaction of the ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 115, pp. 252–261, 1993.

[7] D. N. Nenchev, K. Yoshida, P. Vichitkulsawat, and M. Uchiyama, "Reaction null-space control of flexible structure mounted manipulator systems," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 6, pp. 1011–1023, December 1999.

[8] N. C. Singer and W. P. Seering, "Preshaping command inputs to reduce system vibration," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 112, pp. 76–82, 1990.

[9] I. M. Diaz, E. Pereira, V. Feliu, and J. J. L. Cela, "Concurrent design of multimode input shapers and link dynamics for flexible manipulators," *Mechatronics, IEEE/ASME Transactions on*, vol. PP, no. 99, pp. 1 –6, 2009.

[10] L. Y. Pao, "Input shaping design for flexible systems with multiple actuators," in *Proceedings of the 13th World Congress of the International Federation of Automatic Control*, San Francisco, July 1996.

[11] S. Lim, H. Stevens, and J. P. How, "Input shaping for multi-input flexible systems," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 121, pp. 443–447, September 1999.

[12] J. Fiene and G. Niemeyer, "Toward switching motor control," *Mechatronics, IEEE/ASME Transactions on*, vol. 11, no. 1, pp. 27 – 34, feb. 2006.

[13] E. J. Barth and M. Goldfarb, "A control design method for switching systems with application to pneumatic servo systems," in *2002 ASME International Mechanical Engineering Congress and Exposition (IMECE)*, ASME. New Orleans, LA: ASME, November 2002, p. 33424.

[14] W. Singhose, N. Singer, and W. Seering, "Residual vibration reduction using vector diagrams to generate shaped inputs," *Transactions of the ASME Journal of Mechanical Design*, vol. 116, pp. 654–659, June 1994.

[15] W. E. Singhose, B. W. Mills, and W. P. Seering, "Closed form methods for generating on-off commands for undamped flexible spacecraft," *Journal of Guidance, Control, and Dynamics*, vol. 22, no. 2, pp. 378–382, 1998.

[16] W. Singhose, E. Biediger, H. Okada, and S. Matunaga, "Experimental verification of real-time control for flexible systems with on-off actuators," *ASME Journal of Dynamic Systems, Measurement, and Control*, vol. 128, pp. 287–296, June 2006.

[17] J. Shan, D. Sun, and D. Liu, "Design for robust component synthesis vibration suppression of flexible structures with on-off actuators," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 512–525, June 2004.

[18] G. Song, N. V. Buck, and B. N. Agrawal, "Spacecraft vibration reduction using pulse-width pulse-frequency modulated input shaper," in *AIAA Guidance, Navigation and Control Conference*. Baltimore, MD: AIAA, 1998.

[19] J. Schultz and J. Ueda, "Discrete switching vibration suppression for flexible systems with redundant actuation," in *International Conference on Advanced Intelligent Mechatronics (AIM2009)*. Singapore: ASME/IEEE, July 2009.

[20] R. Newnham, A. Dogan, Q. Xu, K. Onitsuka, J. Tressler, and S. Yoshikawa, "Flextensional "moonie" actuators," in *Ultrasonics Symposium*. IEEE, 1993, pp. 509–513.

[21] J. Ueda, T. Secord, and H. Asada, "Piezoelectric cellular actuators using nested rhombus multilayer mechanisms," in *1st Annual Dynamic Systems and Control Conference (DSCC 2008)*, Ann Arbor, MI, USA, October 2008.

[22] J. Ueda, T. Secord, and H. Asada., "Large effective-strain piezoelectric actuators using nested cellular architecture with exponential strain amplification mechanisms," *IEEE/ASME Transactions on Mechatronics*, vol. 15, pp. 770–782, 2010.

[23] I. CEDRAT. [Online]. Available: http://www.cedrat.com

[24] T. W. Secord and H. H. Asada, "A variable stiffness actuator with tunable resonance for cyclic motion tasks," in *IEEE International Conference on Robotics and Automation*, Kobe, Japan, May 2009, p. 176181.

[25] K. Uchino, *Piezoelectric Actuators and Ultrasonic Motors*, ser. Electronic Materials: Science and Technology, H. L. Tuller, Ed. Kluwer Academic Publishers, 1997.

[26] M. Powell, *Numerical Methods for Nonlinear Algebraic Equations*. Gordon and Breach Science Publishers, 1970, ch. 7, pp. 115–161.

[27] G. Alefeld and J. Hertzberger, *Introduction to Interval Computations*, W. Rheinboldt, Ed. New York: Academic Press, 1983.